Multi-Agent Surveillance

Hendrik Baacke *i6175085,* Pierre Bongrand *i6175237,* Alexandros Chimonas *i6151603,* Olga Fadeitcheva *i6211002,* Maaike Hooghiemstra *i6119992,* Marco Rietjens *i6174997*

***Abstract*— The report investigates how a linear regression model approach can be used to compute interactions in a multi-agent system with two agent types and conflicting objectives. The linear model distinguishes itself by a high scalability and provides a framework which can be easily optimized to specific surveillance simulations. The model is implemented for both the surveillance and the intruder agent with different additional heuristics such as an A* path-finding algorithm for the intruder and a global optimization with a genetic algorithm for the guard. Furthermore, implications of indirect communication via a swarm-intelligence approach and how the agent performance differs in varying types of environment are explored.**

***Index Terms*—Multi-agent system, Surveillance Strategies, Linear Regression, Genetic Algorithm, A*-algorithm, Pheromone Communication, Swarm Intelligence**

I. Introduction

THE concept of multi-agent systems arose in the 1970's [1]. It heavily draws from a broad range of disciplines, among which are artificial intelligence, computer science and game theory. Research and development in the field has been busy during the last years and is being accelerated by an increasing demand for automatization of complex systems. Among the reasons for the higher demand are cost considerations and security concerns for people involved in high-risk tasks. One of these high-risk tasks is to surveil and cover an environment with security guards.

A. Problem definition

In the simulation which is discussed in this report two agent types are essentially trying to outperform each other. The surveillance agents have the the objective to detect intruders and then catch them, using intelligent and coordinated behaviour while also taking their acquired knowledge about the map into consideration [1], [2], [3]. The system needs to implement an intelligent way of dividing tasks and handle communication among robots [1], [2], [3], [4], [5]. In order to succeed, the intruder agents need to use effective evasion tactics along with path-finding to reach the goal of entering the target area.

The problem definition arises from these requirements and can be expressed as:
Implement a multi-agent surveillance and intervention system that uses its available resources, such as vision, hearing and communication, as well as its knowledge about the environment in order to catch intruders at a minimal time. Define an intruder AI which can compete with the surveillance AI using intelligent path-finding and coordination.

B. Research questions

Due to the large number of different components that are an inherent part of the simulation, multiple research questions can be identified which appear to be attractive from an analytical point of view.
The first question concerns how different map structures influence the performance of the surveillance system. This is followed by an analysis of whether and by how much the agent-to-agent communication actually helps the surveillance agents to coordinate for intercepting intruders. Additionally, the question is posed whether genetic optimization of a surveillance agent makes a difference regarding the performance metrics.
The previously mentioned research questions are all posed from the perspective of the surveillance AI. To investigate how the performance of the intruders differ when using the A* path-finding algorithm, a comparison between the intruder algorithm making only use of its evaluation function and the A* assisted version is performed. According to this the research questions are:

1) *How does the proposed surveillance approach perform when introducing varying environmental features?*

2) *Does the multi-agent surveillance system benefit from swarm-based communication?*

3) *How much can the approach be optimized making use of a genetic algorithm?*

4) *How much does the intruder AI benefit from a path-finding algorithm?*

C. Literature review

Extensive research has been conducted in the past on designing systems of multiple interacting agents. The focus of this work lies on how the agents can best coordinate themselves via different communication strategies and how to interact with the environment in a way that area coverage is maximized. A strong influence in that sense has the work of [1]. For the automated simulation of different environments Weiss states that the coordination of activities and the decision making of agents are of great importance [1], [5]. The basis for communication in a multi-agent surveillance framework and how to insert communication acts into the single-agent plans is given by [4] and [3] . If there is a link between different agents, then they should have a direct communication mechanism [5]. Implications of indirect communication, input and communication delays, are explored in [6]. This work has relevance for a pheromone based communication system which is later defined in the methodology section III.

A key concept for surveillance agents is the optimal coverage of the area. This can only be achieved by deploying an intelligent patrolling strategy in order to maximize coverage at any time. Inspiration on how to implement such behaviour is taken from [7].
Milella et al. state that a distributed control architecture can play a huge role in surveilling an environment and intercepting intruders [3]. They also discuss a promising method for memorization of the environment via a Bayesian Belief Network for each agent. Communicating changes of intruder position can be done by setting up each agent's belief network as a node in a graph and relaying information to other nodes, depending on the communication cost within the simulation [6], [3]. A slightly different but nonetheless interesting approach is proposed by [8]. The authors of [8] highlight an adaptive control for monitoring based on positional characteristics of the different agent types. They also present the variation and the dynamics of the system with respect to certain parameters of the system like the participating number of agents and varying environmental factors. The analytical approach of [8] influences the analysis of experimental outcomes in this paper.

The explorations phase is carried out by the exploration agent which is a different type of agent than the surveillance entity. The agent should try to find all kind of structures within the environment. The breadth-first search algorithm generates a path through the map such that the agent takes the shortest path through the map [1]. However, as can be inferred from the problem definition, this report is mainly concerned with the parallel operations of the surveillance phase and therefore leaves intricacies of the exploration phase aside.

As can be seen, research on multi-agent systems for surveillance applications is an active field. Useful heuristics and strategies from the named articles and papers are extracted and implemented for the purposes of the AI system which is presented in this paper. To put the work conducted and presented in this report into the context of current relevant literature, it can be noted that no ground braking new algorithmic approach for surveillance agents is proposed, but that it rather pursues a practical approach and explores a heuristic strategy for surveilling a map and intercepting agents efficiently by using different means of communication, a range of parameter features and specific weighting of these as well as sophisticated agent vision.

D. Report structure

This report has the aspiration to answer the research questions and does so by first defining the assumptions about the basic properties of the implemented agent types and the environment in which they operate in section (II) (only the relevant assumptions that deviate from the prerequisites). This is followed by an overview of the main functionality of each algorithm, provided in the 'Agent Methodology' section (III). After all, the performance of the different methods must be evaluated and compared. In order to do so, experiments are conducted. Their setup and predictions about the outcome are described in the 'Experiments' section IV which is followed by presenting the results (V). The performance of the bots is then analyzed in the 'Discussion' (VI), in which the main objective is then to get a clear answer for the posed research questions. However, if no satisfying answer can be given, the findings must be interpreted carefully. Finally, agent specific properties and suggestions for possible use in real-world scenarios are highlighted in the 'Conclusion' section (VII).

II. Assumptions

Most of the properties of the agents and the environment are inherited from the project prerequisites and can readily be looked up. However, some properties have been changed. These deviations must be divided into environment specific and agent specific. They are elaborated on in the upcoming subsections. The winning conditions for both the surveillance as well as the intruder agent are the same as in the prerequisites.

A. Environment assumptions

The map, on which the agents operate, is continuous and has a minimum size of 50x50 meters. Four different kind of areas can be created on the map, structures, shade areas, target ares and sentry towers. Structures, shade areas as well as target areas need to be at least 2x2 meters, while sentry towers have a predefined size. No areas may overlap. Depending on the size of the structures, there is a minimum of one door and zero windows and a maximum of six doors and four windows. These are allocated randomly upon the creation of the structure without any overlapping.

B. Agent assumptions

1) Visual: Instead of agents having a visual range, all entities in the world have a visibility range attribute, so the range from which they can be seen. The guards can be seen from 7.5 meters, intruders from 6.0, structures and shades from 12.0, and sentry towers from 24.0.
As soon as a surveillance agent sees one part of a building structure, it has full knowledge about all properties of the building, ie. about its corner points and how many doors and windows there are.

2) Sound: The loudness of the sounds agents create, ie. the radius from which it can be heard when walking, is based on the function y = 4x, with x being the velocity of the agent.

3) General: There is no way for intruders to leave the map, they are spawned randomly around the edge of the map at the beginning of the simulation, while guards are spawned randomly on any point in the map outside buildings. Moreover there exist different pheromone types which serve as a marker for agents. The pheromones grow to a 5 meter radius over a time of 2 seconds.

III. Agent Methodology

This section describes the surveillance algorithms and the intruder algorithms.

A. Surveillance AI

The section describes the underlying mathematical framework of the implemented artificial intelligence and its surveillance tactic. Furthermore, an indirect agent to agent communication approach using pheromones is described. In the end, an optimization strategy in form of a genetic algorithm is introduced.

1) Surveillance approach: The proposed surveillance algorithm, furthermore referred to as 'linear regression heuristic algorithm' (LRHA) is based on the approach proposed by [9], [10] and applies a heuristic evaluation function with a number of decision variables that serve useful features for the agent to achieve its goal of locating the intruder. This approach is summarized in Fig. (1).

The decision variables consist of: visible guards, visible intruders, sounds heard, each type of visible area, the pheromones and the map border. The magnitude of how much each of these variables influences the next direction of agent movement is determined by their respective coefficient in the (**c**)-vector, which has obvious implications on possible optimizing strategies for the algorithm. If the decision variable is an element that needs to be avoided, the coefficient corresponding to that vector is negative, otherwise the coefficient is positive. The dimension of the coefficient vector is not fixed and can be changed to allow for a different decision policy.

Linear regression is used to calculate an observation matrix of 360 x *n* which can be expressed as

$$\mathbf{Ac} = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & & & \\ \vdots & & & \\ a_{360,1} & & & \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} d_{w1} \\ d_{w2} \\ \vdots \\ d_{w360} \end{bmatrix} \tag{1}$$

where *n* denotes the number of decision variables that determine in which direction the agent should go next. As shown in (1), the amount of columns of the observation matrix is defined by the dimension of the coefficient vector. By multiplying the 360 x *n* observation matrix **A** with the coefficient vector **c** of dim *n*, the 360 x 1 decision vector **d** is obtained. The index of the largest value in **d** corresponds to the direction of agent movement in degrees for the next frame of the simulation. The computation is performed repeatedly and for each surveillance agent independently which characterizes a decentralized control architecture.

For example, if a guard spots an intruder in a certain angle θ, the value $a_{\theta,y}$ in the observation matrix, at the column *y* which is to be determined by the index of the decision variable for 'visible intruders' in the coefficient vector, is changed accordingly.

However, this is a simplification of the value assignment. The features, visible guards, visible borders, visible walls, traceable pheromones and sounds, scores are normally distributed around the angle where they are perceived by the agent. This is implemented with the Gaussian normal distribution

$$\frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\theta)^2 / 2\sigma^2} \tag{2}$$

where the expectancy value θ is the direction of the trigger signal in degrees (from 0 to 359) and x are the angles close to θ. The process is designed such that if a surveillance agent perceives another surveillance agent in its proximity, it is suppressed to not only move towards its exact position but also to angles close to its position. The underlying reasoning is that it would be redundant to have two agents covering the same area.

When computing the optimal direction for the next simulation frame, past decisions are to be taken into account. For example when a surveillance agent is pursuing an intruder and the intruder sprints out of visual range, the agent needs to keep a memory of which direction the intruder went in order to make the decision vector more robust. For the beginning frame of the simulation, the first decision vector d_0 is initialized such that the values are assigned randomly from a closed interval between 0 and 5. Therefore,

$$d_0 = \begin{bmatrix} d_{w1 \leftarrow random[0,5]} \\ d_{w2 \leftarrow random[0,5]} \\ \vdots \\ d_{w360 \leftarrow random[0,5]} \end{bmatrix} \tag{3}$$

The computation for each iteration after that is given by

$$\begin{aligned} d_1 &= \mathbf{A}_1 c + \epsilon\, d_0, \\ d_2 &= \mathbf{A}_2 c + \epsilon\, d_1, \\ &\vdots \\ d_n &= \mathbf{A}_n c + \epsilon\, d_{n-1} \end{aligned} \tag{4}$$

where ϵ is the memorization parameter. ϵ is set heuristically to 0.95 and denotes how much the previous decision vector is weighed into the current decision.
(4) can be generalized and written as

$$\begin{aligned} d_n &= \mathbf{A}_n c + \sum_{i=0}^{n-1} \epsilon^{\,n-i} \mathbf{A}_i c \\ &= c(\mathbf{A}_n + \sum_{i=0}^{n-1} \epsilon^{\,n-i}\, \mathbf{A}_i) \end{aligned} \tag{5}$$

By close inspection of (5), it firstly becomes clear that the algorithm prioritizes recent events over events further in the past. Secondly, small values for a coefficient in d_n dissipate after a significantly lower number of frames than higher values. This 'weighted memorization' prioritizes important events and keeps them longer in the memory than those which are lower ranked by the evaluation function (1).

2) Surveillance agent communication: LRHA can implement indirect agent-to-agent communication with 'pheromones'. A swarm intelligence approach is used

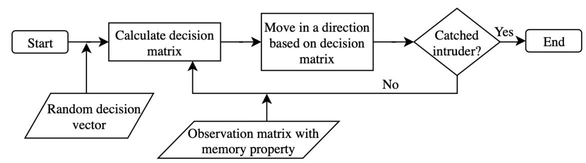

Fig. 1. The state diagram of LRHA.

for this. Specifically an ant colony approach described by [11], [12]. Every agent is able to emit every type of pheromone. Agents discharge pheromones when they see an intruder, when they enter a shading area and upon hearing a sound. However, this assignment is non-static, meaning that new pheromones for different actions can easily be added. Pheromone emission is tied to a cooldown period.

Pheromones have a growth rate and a maximum radius. Moreover, they spread uniformly from the position of emission to their maximum radius. Pheromones are a part of the evaluation function which means if an agent spots a pheromone, its observation matrix is changed accordingly. This has an impact on the decision vector of the next frame. However, direct encounters, like spotting an intruder, are always given preference over the pheromones by setting heuristic weights in the coefficient vector and assigning lower scores to the observation matrix. The LRHA can function with or without communication between agents.

How the performance changes when the communication is disabled is part of the open research question and remains to be discovered in the analysis part.

3) Genetic algorithm for global optimization implemented for the heuristic approach: The genetic algorithm, as described in depth in [13], is used in addition with the heuristic bot to achieve a global optimization. The resulting AI is furthermore referred to as OLRA, 'optimized linear regression algorithm'. The element that should be optimized within the heuristic bot is the coefficient variable vector **c** from (1). In order to do so, a certain number of different coefficient vectors are initially generated randomly. Afterwards, a predefined amount of simulations is applied to each of the different **c**-vectors . The number of times the guards won during these simulations is used as the fitness variable. The combination of **c** as well as the fitness variable is given as an input to the genetic algorithm in order to generate a new population which in this case translates to a new set of coefficient vectors. The method used to generate a new population is as follows: 20 percent of the newly generated coefficient vectors are generated by a crossover between the best **c** and a randomly chosen one from the other elite set, 20 percent are generated by a crossover between the second best and a randomly chosen **c** from the other elite ones, same goes for the third best and 20% is generated by a random crossover between two of the elite coefficient vectors. The last 20% is chosen randomly from the elite set, which is the set of the best coefficient vectors. The crossover function is defined by

$$z_i^1 = \alpha x_i + (1 - \alpha) y_i \tag{6}$$

where $\alpha \in [-0.5, 1.5]$ is taken randomly for each coordinate and x and y are the **c**'s chosen from the elite set as described above. The results of this crossover function are then, under a certain probability, mixed with a mutation. The elite set is chosen by selecting the best **c**'s from the **c**'s that have been used during the simulations. It's size can vary depending on the elite size rate given. At each iteration, the newly calculated population is applied in the new simulations, and new results are generated providing a better fitness variable for each and one of the new **c**'s. The iterations stop when the **c**'s converge. Convergence in this case means that the **c**'s have a really small, or none at all, variation from one iteration to another.

B. Intruder AI

The section takes up the idea of the linear regression model and develops it further for the application in an intruder artificial intelligence. After this, an A* path-finding algorithm for improving the performance of the bot is described.

1) Intruder approach: The linear regression model which is implemented for the guard, is also the framework for the intruder AI. The key difference between the two is that most decision variables, represented by the **d**-vector in(1) are different. For example, the fact that the intruders do not make use of any indirect communication leads to all decision variables associated to pheromones not being present in the observation matrix **A** in (1). In turn, the target area is added as a feature with a large positive coefficient. Some decision variables, like the number of visible surveillance agents stay in the model, but the coefficients in the **c**-vector are set to different values. Intruders want to avoid guards and themselves to not draw too much attention to one spot, so the coefficients for these decision variables are negative. Furthermore, some additional heuristics are used by the intruder according to the values in **A** for a given frame (5). For example, the intruder starts to sprint as soon as a guard is spotted.

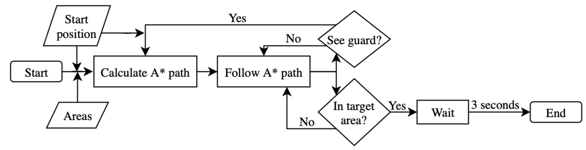

Fig. 2. The state diagram of the A* algorithm implemented for the intruder agent.

2) A-algorithm for path finding:* Contrary to the guards, the intruders have the well-defined discrete objective to enter the target area until the winning condition is met. To navigate the map efficiently, an A*-algorithm is used which is defined in [14]. Its state diagram is given by Fig. (2) . For this application, a graph-based approach is chosen. As illustrated in Fig. (3, in order to create the graph, a close position next to the corners of the structures and sentry towers are set as nodes, as well as the center of the shade areas and the target area. The initial position of the intruder is also a node, and evidently the starting position. Before running A*, all of the nodes that can be connected (not being intercepted by a structure) are connected and create an edge. A weight is then assigned to each edge of the graph. The heuristic used to determine this weight is simply the euclidean distance. At the first iteration, the A star algorithm is calculated to determine the best path to arrive from the bots starting position to the target area. However, if an intruder sees a guard in front of him, the edges that surround the guard get assigned a bigger weight, and the A*-algorithm is once again calculated to provide the intruder a new path to follow.

IV. Experiments

The section describes the performance metrics and the conducted experiments. Along with the description of the experimental setup, a brief conjecture is provided which highlights the expected outcome. After each experiment is run 1000 times and results are obtained in (V), a statistical analysis is performed and the outcomes, which are relevant for answering the research questions, are provided in (VI).

A. Performance metrics

It is necessary that before running any experiment regarding the comparative analysis of the AI's or the impact of environmental parameters it needs to be defined how performance is measured and regarding to which

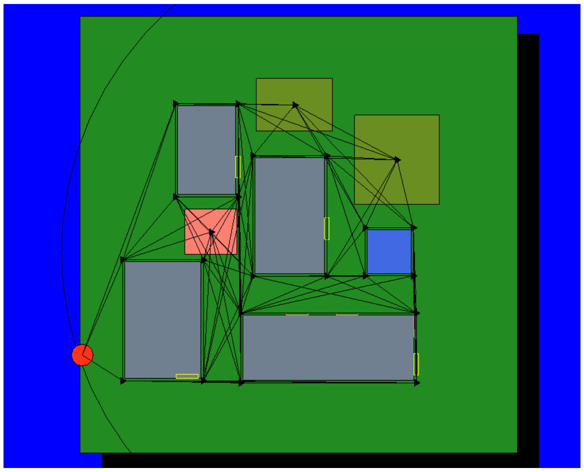

Fig. 3. The graph for A* spans vertices close to all four corner points on the outside of structures (grey) and sentry towers (blue) and in the center of the target (red) and shade areas (green).

criteria a performance is analyzed.
The win-rate of an algorithm provides a robust measure of success. Additional to that, the average time needed to catch an intruder or how long it takes to reach the target area, depending from which perspective the experiment is posed (guard- or intruder AI), is used as a metric.

B. Environment experiments

The set of experiments is targeted at the first research question, mentioned in section (I-B). To answer this question, the position and size of the target area as well as the density of the buildings are changed. All environment experiments are designed in such a way that the surveillance bot operates with 15 agents against 5 intruders which make use of A*.

1) Experiment 1 - Influence of target position and target size: The experiment for target area position is performed on five maps which have the target area centered and 5 maps that have the target area close to the edge of the map. The simulation is run multiple times in order to achieve a robust result.
Two maps are created for the target size and the same process as mentioned before is applied.
It is expected that since the intruders spawn randomly at the edge of the map, moving the target area further from the center of the map has a negative impact on the guards performance and enables the intruder to win the simulation faster. Moreover, if the target area is increased in size, the intruder win-rate is expected to increase.

2) Experiment 2 - Influence of structure density: The experiment is designed such that the surveillance agent operates against the intruder AI on three different map density classes plus a 'labyrinth-like' map class. By looking at how the algorithm for the surveillance agent is implemented, it can be predicted that the more labyrinth-like the map is, the poorer the surveillance agents perform. This is because of the very efficient A* path-finding that the intruder agents implement.

C. Communication experiments

The set of experiments is targeted at the second research question, mentioned in section (I-B). The communication experiments are designed in such a way that the surveillance bot operates with 15 agents against 5 intruders which make use of A*.

1) Experiment 3 - Disabling the pheromone communication of the surveillance AI: For this experiment, a map with average structure density, similar to a possible real world environment, is used. The surveillance agents using communication are compared to the form where the pheromone communication is disabled. Upon suppressing the pheromone communication, it is expected that the surveillance agents perform worse. Using the evaluation function solely based on direct visual or audio input has disadvantages because of the smaller receptive area compared to when pheromones are enabled.

D. Optimization experiment

1) Experiment 4 - Comparing vanilla LRHA to a genetically optimized version: This experiment is concerned with the third research question of section (I-B). Because the genetic algorithm optimizes for one specific map layout and guard/intruder combination, these parameters are fixed. As for the previous experiment, a 'realistic' map is chosen and there are 15 guards and 5 intruders in the simulation.

The performance of an optimized algorithm is put in relation to a basic version of the algorithm. It is expected that the optimized version does perform better on the map it was optimized for, because its coefficient vector been boosted to fit a specific simulation, whereas the vanilla agent implements the heuristically chosen coefficient vector.

E. Intruder tactic experiment

1) Experiment 5 - Comparing intruder performance with and without A-path finding:* This experiment is targeted at the fourth research question and is designed with 5 intruders against 15 guards. It is run on three maps. The first has an average structure density, the second one has a labyrinth-like structure, whereas the third one has a standard structure density with shades and sentry towers. The experiment is performed to highlight the difference in performance of a vanilla intruder, solely relying on its evaluation function, and a more sophisticated algorithm which is going to the target area following a calculated path. It is to be expected that the intruder algorithm with path-finding works better in all cases than the normal algorithm. However, the difference should be more obvious for the labyrinth map than for the average map.

V. Results

This section presents the results obtained from the experiments that are described in the previous section. It also gives a statistical analysis on the results and gives explanatory graphs and figures.

A. Results experiment 1

exp1 The results obtained for the first experiment:

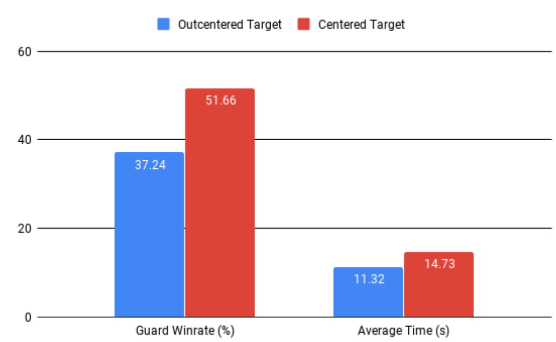

Fig. 4. The mean performance of the surveillance agent upon changing the target area position.

Fig. V-A) presents the win-rate of the guards and the average time it takes to catch an intruder. There is a high difference in win-rates of maps with non-centered target area versus centered target areas. On the contrary there is a small difference in average time.To actually see if there is a significant difference, the binomial test is used for comparison of win-rates and the Welch's approach is used for comparison of average time.

Due to the large sample size, the normal distribution is used to solve the binomial distribution for two samples. This can be used because in all cases $n \times p$ and $n \times q$ are both larger than 5 [15]. The null hypothesis is that the win-rate of maps that have a non-centered target area does not differ from the win-rate of maps that have a centered target area. Appendix VII-A presents the results on this test. It can be seen that for a relatively large map (size '4') the difference is not significant, so the null hypothesis cannot be rejected. For all the other map sizes, the difference is significant, so the null hypothesis is rejected. Hence, in most cases the null hypothesis is rejected.

The null hypothesis for comparing average time to catch an intruder is that this should not differ for maps, of all sizes, with a non-centered target area and maps with a centered target area. Appendix VII-A also provides the Welch's approach that has been done [16]. It can be seen that the difference is not significant, so the null hypothesis cannot be rejected.

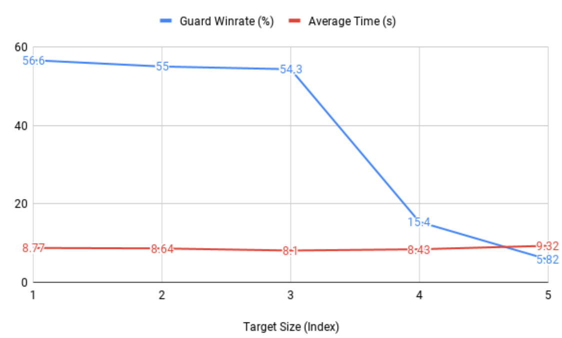

Fig. 5. The mean performance of the surveillance agent upon changing the target size from small (1) to large (5).

Fig. (V-A) shows that the guard win rate stays stable for small to medium target sizes and then drops from around 55% to only 15% for size for and further to 5% for target size 5. The average time stays constant throughout the experiment.

B. Results experiment 2

Fig. (V-B) shows the guard's win-rate upon operating in environments of different building densities. Building density refers to the percentage of map covered with building structures. The covered area can be divided up by a different number of buildings, for example a map with one large building may have the same building density as a map with two smaller buildings. The figure also illustrates that LRHA appears to perform a lot worse in maps with more and smaller buildings, like labyrinth maps. The performance on maps of low building density do not follow the trend. The performance in these cases stays constant.

The difference on time to catch the intruder between

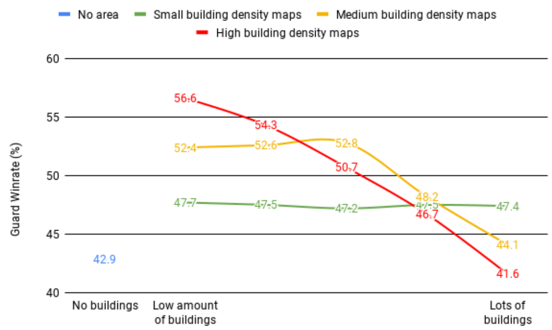

Fig. 6. Surveillance agent performance upon changing the amount and proportion of building structures in a map.

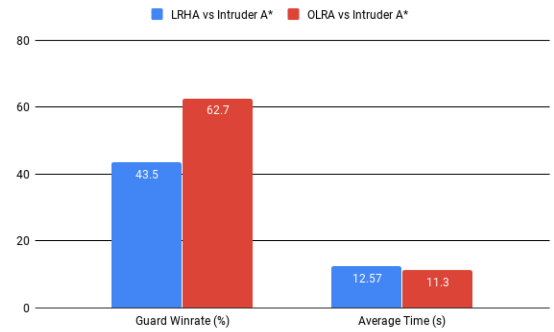

Fig. 8. A performance comparison of OLRA, an optimized algorithm and LRHA, the base algorithm regarding an average simulation environment.

maps having a 'normal' target size (size '3') against other target sizes is tested with the Welch's test [16]. The null hypothesis is that the there is no difference between the average times. From appendix VII-B it can be seen that zero is not inside all the confidence intervals, so the difference is significant. Hence, the null hypothesis stated above is rejected.

C. Results experiment 3

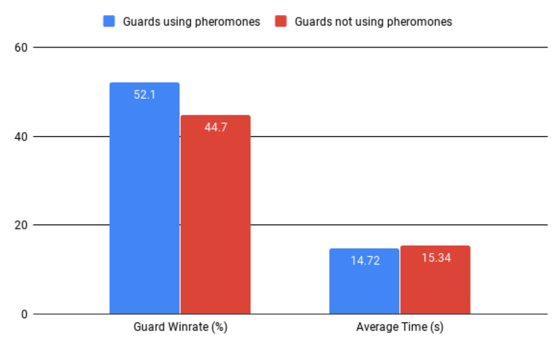

Fig. 7. A performance comparison between LRHA with and without pheromone communication.

Fig. (V-C) shows that the difference of win rate between the guards that are using pheromones and the ones who are not is about 7.5%. The difference of average time between the two different strategies is small. However, it can be noted that also regarding this performance metric, the bot which implements communication is at an advantage. This advantage is quantifiable by 0.5 seconds.

D. Results experiment 4

Fig. (V-D) illustrates that an algorithm with a boosted coefficient vector which fits the environment properties optimally outperforms the vanilla approach by some margin. In this particular case, the OLRA outperforms the LRHA by almost 20% in terms of winning rate. Nevertheless, the average times to capture the intruder are quite similar from OLRA to LRHA, giving a advantage of 1 second to OLRA.

E. Results experiment 5

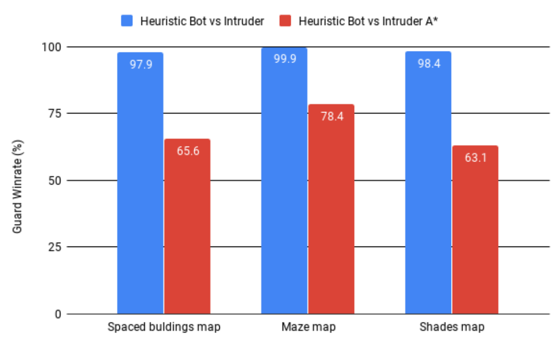

Fig. 9. Comparing how LRHA performs against intruders with and without A* on 3 different map layouts

It can be seen from Fig. (V-E) that LRHA always outperforms the simple intruder algorithm which uses the same linear model, non regarding the map layout. The performance against the smart intruder is worse, however still on a high level. This is due to the chosen maps for testing.

VI. Discussion

The section concludes on statistical analysis of the results given in the previous section and aspires to provide an answer to each research question based on this.

A. Performance of a surveillance agent in different environments

Changing the target area position from the center to the outside region of a map implies that the area moves closer to the spawning points of the intruders which explains the obtained result. The results in subsection (IV-B1) make clear that for most map-cases the win rate differs for non-centered target areas compared to centered target areas. If a target area is non-centered, so for example in the top right corner, then because the intruder gets spawned on the border, on average it's linear distance to the target area decreases, thus also decreasing the chance of intersecting a guard. In this case, the intruder also does not encounter as many buildings, which in turn makes makes it easier again. On the contrary, the average time to catch an intruder does not differ for the same environmental changes as stated before. Since guards get spawned randomly, the distribution of their distance to the intruders does not change.
Moreover, from the results subsection (V-B) it can be seen that the null hypothesis, namely that there is no difference in time to catch the intruder between having a 'normal' target size against having an other target size, is rejected. This means that the average time to catch the intruder does differ when having different target sizes. This can be explained by the fact that it is easier for the intruder to 'win' if the target size is bigger than normal. In the same way, it can be explained that it is more difficult for the intruder when having a smaller target size. Hence, it can be stated that changing the map properties does not impact the performance of a surveillance AI using LRHA as significantly as it impacts the performance of an intruder AI using LRHA.

B. Genetic algorithm on optimization

Through help of the fourth experiment, it can be seen that an optimization algorithm is helping the LRHA to obtain better results. The training of the **c**-vector through the genetic algorithm, which takes large time and computation resources, provided a globally optimized solution. However, this solution is only adapted to the map on which the algorithm was trained. Further investigations would need to be performed in order to establish how well the OLRA would perform on other similar maps and on irregular environments ie. low/high building density, low/high shade density, few sentry towers.

C. Benefits of a path-finding algorithm for intruders

The fifth experiment demonstrates how the A* algorithm implemented for the intruder bot has a negative impact on the performance of the surveillance agents. The experiment is run on 3 different maps. In each case, a surveillance AI using LRHA was put against an intruder AI using LRHA, and another time against an intruder AI using A*. In each case the guards were put against the intruders using the A* algorithm, the performance rate dropped at least by 10%. This means that the usage of a path-finding algorithm on the intruders does have an impact on the surveillance agents performance.
This increase of intelligence can be explained by the fact that the intruders have full knowledge of the map and of the target area's position. The way the A* algorithm is implemented, it could not have been done without this prior knowledge. Therefore, this prior knowledge about the environment makes a big difference. While the A* star algorithm calculates the shortest and most optimal path on a known map, the guards are trying to find and catch an intruder which is a more complicated task. The fact that the intruders have this knowledge but not the guards, has a big impact on the overall results. Furthermore, the fact that the intruders can take sprint back as soon as they see a guard, calculate a new path taking in to account the guard they just saw, gives them multiple attempts, thus improving their overall results.

D. Benefits of Swarm-Based communication

To answer the research question concerning the impact of a swarm-based communication, the results of the third experiment provide surprising results. As it can be seen in Fig. (V-C), the guards winning rate is 52.1% when using pheromone based communication, and only decreases by 7.4% to obtain a rate of 44.7% when not using any indirect communication. On top of that, the average time does not significantly vary between both of the experiments. These results do not align with the expectations because the usage of pheromones to transmit messages in order to enhance the surveillance performance, barely has any measurable impact. This shows that other coefficients in the **c**-vector, such as the visibility of intruders or guards, have a stronger influence on the bots decisions than the pheromones. Furthermore, in the experiments the meaning assigned to each of the pheromones consists of: red, in case a guard sees an intruder, yellow if a guard is in a sentry tower, blue if they hear a noise, purple if a guard sees two other guards and green if a guard senses a red pheromone. It is highly probable that the information that was chosen to be shared through indirect communication simply is not expressive enough. Other meanings should be assigned to each pheromone in order to see how whether performance improves.

VII. Conclusion

To give a conclusive answer to the problem statement, the research questions are going to be answered with the help of the results provided.

The results of the experiments performed leads to identify how well balanced maps can be defined. Such maps have a centered target area of an average size, with a standard building density that can take different shape such as labyrinth. It is shown that in those well balanced environments the LRHA adapts and displays a strong performance.

Surprisingly, the swarm-based communication did not function as predicted. The usage of an indirect communication through pheromones does not provide significant differences in comparison to a simulation without communication.

The OLRA is proven to be functional in order to provide a global optimization to the LRHA's coefficient vector. Further research can be done by applying the genetic algorithm on different types of maps, in order to see how the coefficient vector from the LRHA varies depending on the environment.

On the other hand, the A* algorithm implemented specifically for the intruder bots yields positive results. This path-finding algorithm proves to work well against the LRHA algorithm which is used on guards.

References

[1] G. Weiss, "Multiagent systems: a modern approach to distributed artificial intelligence," *INTERNATIONAL JOURNAL OF COMPUTATIONAL INTELLIGENCE AND APPLICATIONS*, vol. 1, pp. 331–334, 2001.

[2] A. M. Uhrmacher and D. Weyns, *Multi-Agent Systems: Simulation and Applications*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 2009.

[3] A. Milella, D. Di Paola, P. L. Mazzeo, P. Spagnolo, M. Leo, G. Cicirelli, and T. D'Orazio, "Active surveillance of dynamic environments using a multi-agent system," vol. 7, 01 2010.

[4] M. Georgeff, "Communication and interaction in multi-agent planning**this research was supported in part by onr contract n000014-80-c-0296 and in part by afosr contract f49620-79-c-0188." in *Readings in Distributed Artificial Intelligence*, A. H. Bond and L. Gasser, Eds. Morgan Kaufmann, 1988, pp. 200 – 204. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B978093461363750022X

[5] D. Vallejo, J. Albusac, J. Castro-Schez, C. Glez-Morcillo, and L. Jimnez, "A multi-agent architecture for supporting distributed normality-based intelligent surveillance," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 2, pp. 325 – 340, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197610002216

[6] Y. Tian and C. Liu, "Consensus of multi-agent systems with diverse input and communication delays," *IEEE Transactions on Automatic Control*, vol. 53, no. 9, pp. 2122–2128, Oct 2008.

[7] Y. Chevaleyre, F. Sempe, and G. Ramalho, "A theoretical analysis of multi-agent patrolling strategies," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.*, July 2004, pp. 1524–1525.

[8] C. Sur and A. Shukla, "Coalition formation for multi-agent coordination for surveillance and capture of foreign intruder using bacteria foraging algorithm," *IFAC Proceedings Volumes*, vol. 47, no. 1, pp. 511 – 518, 2014, 3rd International Conference on Advances in Control and Optimization of Dynamical Systems (2014). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016327033

[9] G. Kuhlmann, K. Dresner, and P. Stone, "Automatic heuristic construction in a complete general game player," in *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, ser. AAAI'06. AAAI Press, 2006, pp. 1457–1462. [Online]. Available: http://dl.acm.org/citation.cfm?id=1597348.1597420

[10] J. E. Clune, "Heuristic evaluation functions for general game playing," *KI - Künstliche Intelligenz*, vol. 25, no. 1, pp. 73–74, Mar 2011. [Online]. Available: https://doi.org/10.1007/s13218-010-0074-7

[11] H. Van Dyke Parunak, ""go to the ant": Engineering principles from natural multi-agent systems," *Annals of Operations Research*, vol. 75, no. 0, pp. 69–101, Jan 1997. [Online]. Available: https://doi.org/10.1023/A:1018980001403

[12] A. Drogoul and J. Ferber, "Multi-agent simulation as a tool for modeling societies: Application to social differentiation in ant colonies," in *Artificial Social Systems*, C. Castelfranchi and E. Werner, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 2–23.

[13] C. Toledo, L. Oliveira, and P. Frana, "Global optimization using a genetic algorithm with hierarchically structured population," *Journal of Computational and Applied Mathematics*, vol. 261, pp. 341 – 351, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377042713006274

[14] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, July 1968.

[15] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, *Probability and statistics for engineers and scientists*. Macmillan New York, 1993, vol. 5.
[16] G. D. Ruxton, "The unequal variance t-test is an underused alternative to student's t-test and the mann–whitney u test," *Behavioral Ecology*, vol. 17, no. 4, pp. 688–690, 2006.

APPENDICES

A. *Appendix A - Contribution*

	H.Baacke	P.Bongrand	A.Chimonas	O.Fadeitcheva	M.Hooghiemstra	M.Rietjens
Report	5	4	2	4	4	2
Statistical Analysis	3	4	2	1	5	2
Research	5	5	3	3	3	3
Presentation	4	5	2	3	4	2
Maps	4	5	3	3	3	3
GUI/Interface	3	1	3	3	4	5
Game Structure	2	3	4	4	3	5
Agent Capabilities	3	2	5	3	4	5
World Attributes	3	2	3	4	3	4
Linear Regresion Heuristic Bot	3	3	5	5	1	3
A Star	2	3	3	5	2	2
Direct Communication	1	2	1	2	3	5
Pheromone Communication	4	3	5	5	2	5
Genetic Algorithm	2	3	3	5	1	1
Exploration algorithm	3	3	2	3	5	4

Fig. 10. Table that represents the attribution of points that each team member assigns of his own contribution to the project on a scale from 1 to 5 per task.

B. *Appendix B - Results for experiment 1*

1 - Target Area Position						
DATA				STATISTICAL ANALYSIS		
Number of experiment to perform: 10						
	Winrates				Winrates	
Maps	Outside	Inside			Outside	Inside
1	0.33	0.514		Mean	0.3724	0.5166
2	0.344	0.511		Standard Deviation	0.05788609505	0.04547856638
3	0.324	0.542		95% Confidence +/-	0.05073846714	0.03986298858
4	0.406	0.447		Lower Bound	0.3216615329	0.4767370114
5	0.458	0.569		Higher Bound	0.4231384671	0.5564629886
Binominal confidence interval	phead	phead1-phead2	Lower bound	Upperbound	Significant	
	0.422	-0.184	-0.226532632	-0.1414673677	Yes	
	0.4275	-0.167	-0.209741618	-0.1242583811	Yes	
	0.433	-0.218	-0.260367747	-0.1756322526	Yes	
	0.4265	-0.041	-0.084313561	0.002313561017	No	
	0.5135	-0.111	-0.154540022	-0.06745997749	Yes	
	Average Time				Average Time	
Maps	Outside	Inside			Outside	Inside
1	12.009374	10.542608		Mean	11.3224592	14.73357
2	11.311388	10.630857		Standard Deviation	1.081926804	9.307198791
3	12.418699	8.514323		95% Confidence +/-	0.9483332321	8.157969537
4	11.284476	12.821937		Lower Bound	10.37412597	6.575600463
5	9.588359	31.158125		Higher Bound	12.27079243	22.89153954
				Welch's confidence interval:	alpha	0.05
				f-value	70.96022811	70
	Outcentered Target	Centered Target		Test statistic --> when data is done	1.995	
Guard Winrate (%)	37.24	51.66		Interval	-11.77082807	4.948606472
Average Time (s)	11.32	14.73		Significant	No	

Fig. 11. Table with raw result data from the first experiment on target position.

C. *Appendix C - Results for experiment 1*

D. *Appendix D - Results for experiment 2*

Simulation	1000			
Target size	**Mean Time**	**Variance Time**		
1	8.765261	2.104738		
2	8.639091	1.927493		
3	8.101718	3.629463		
4	8.4301	1.203948		
5	9.328215	5.392845		
All compare with size 3 (normal size)				
	Normal distribution			
fvalue	**tstatistic**	Lowerbound	Upperbound	Significant?
1866.063875	1.96	0.5151229658	0.8119630342	Yes
1826.649597	1.96	0.3912648135	0.6834811865	Yes
1596.06853	1.96	0.1921174349	0.4646465651	Yes
1924.485639	1.96	1.040324772	1.412669228	Yes

Fig. 12. Table with raw result data from the first experiment on target size.

Building Density				
DATA				
Number of experiment to perform: 10				
Winrates				
#Buildings/ Area	No area	Small building density maps	Medium building density maps	High building density maps
No buildings	42.9			
Low amount of buildings		47.7	52.4	56.6
		47.5	52.6	54.3
		47.2	52.8	50.7
		47.5	48.2	46.7
Lots of buildings		47.4	44.1	41.6
#Buildings/ Area	No area	Small Area	Medium Area	High Area
No Buildings	4.368534			
Few Buildings		4.5944095	5.1645594	5.282743
Medium Buildings		5.482768	4.861676	5.5019064
Lots of Buildings		5.7907863	5.009222	15.508804

Fig. 13. Table with raw result data from the second experiment on building density.